HRM Control System Features for HINS in an EPICS Environment

Robert Goodwin

Thu, Nov 2, 2006

A front end uses a PowerPC cpu board in a VME crate plus one or two SPAN boards to expand the number of PMC slots supported to, at most, 6. Since the front end software itself needs 2 PMC slots, up to 4 slots can be made available to HRMs. Each HRM connects to one PMC slot on a SPAN board via a high-speed (320Mbit) serial link.

Inside the HRM itself are one or two 64-channel "Slow Data" digitizers. Each digitizes 64 channels every 100 μs, automatically delivering the results via the link into a 2MB RAM area on the PMC board in the VME crate, using about 10% of the available link bandwidth. The 2MB circular buffer memory allows for 16K copies of 64 channels of digitized values that "wrap" in about 1.6 seconds. A register available on the PMC board allows monitoring which of the 16K copies is currently being filled, so that the cpu can find the latest readings.

Also included in the HRM, besides 64–128 channels of A/D, are 8 D/A's, 8 bytes of Digital I/O, and 8 timers. The cpu sees these as registers it can read that reflect what is in the HRM. The cpu can write to these registers to change what is in the HRM. The serial communication is automatic.

The front end performs the same work every 10 Hz cycle, triggered by a clock event plus delay, so that its operation is in sync with the accelerator hardware. At first, it updates its own data pool with the latest analog and digital readings. It calls each enabled local application to give it a chance to run (briefly). It then returns replies for any active data request for which a reply is due on that cycle. It checks for alarm conditions for all active analog and digital values. (This may be handled by EPICS.) It gives a chance for the currently active page application to run. Then, for the most part, it is idle for the rest of the cycle.

Network protocols supported by the front end software are of two types, each based on UDP. The Classic protocol is the original one supported. It is used by a suite of page applications that can run within the front end to aid in maintenance and configuration of these systems. Acnet protocols are also supported but not required for operation of the front end.

The EPICS connection can be done in two ways. The one currently used is to load EPICS software into the same vxWorks-based front end system. Device support routines ask for data of interest for user displays by calling an access routine that samples the latest readings from the data pool.

A second approach is to maintain a separation between an EPICS front end (IOC) and the front end described here by using a different cpu for each. This could aid debugging. Access to front end data would be made via the network, probably using Classic protocol.

Access to waveform data depends upon the digitizer used. For slow signals, for which 10KHz sampling is sufficient, a Classic request can be made to deliver replies that continuously sample the HRM Slow Data circular buffer to keep up with a desired rate. Both time and data are included for each point, with the time values relative to a selected clock event.

For faster waveforms, a higher speed digitizer is needed that can deposit the digitized data points into a simple array of data readings. A Classic request can access this waveform buffer.